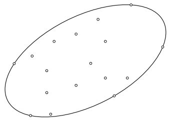
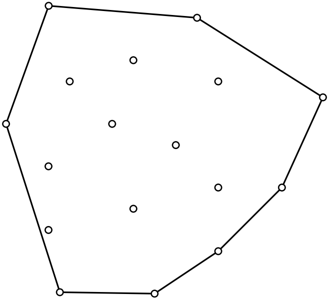
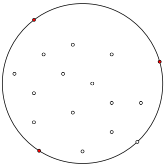
Vorlesung Algorithmen und Wahrscheinlichkeit, D-INFK, ETH Zürich

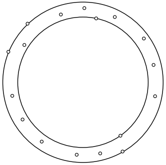
Angelika Steger & Emo Welzl

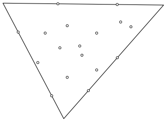
Konvexe Hülle: Jarvis Wrap

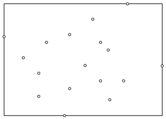


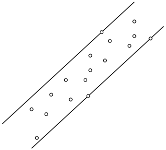












Problemstellung

ConvexHull-Problem. Gegeben eine endliche Punktemenge $P \subseteq \mathbb{R}^2$, bestimme die konvexe Hülle von P.

Problemstellung

ConvexHull-Problem. Gegeben eine endliche Punktemenge $P \subseteq \mathbb{R}^2$, bestimme die konvexe Hülle von P.

- Was ist die konvexe Hülle?
- Wie stellt man die konvexe Hülle dar?

Problemstellung

ConvexHull-Problem. Gegeben eine endliche Punktemenge $P \subseteq \mathbb{R}^2$, bestimme die konvexe Hülle von P.

- Was ist die konvexe Hülle?
- Wie stellt man die konvexe Hülle dar?

- Wie geht man mit Spezialfällen und numerischen Problemen um?

Konvexe Menge, konvexe Hülle

Sei $d \in \mathbb{N}$.

- Für $v_0, v_1 \in \mathbb{R}^d$ sei

$$\overline{v_0 v_1} := \{(1-\lambda)v_0 + \lambda v_1 \mid \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1\} ,$$

  das v_0 und v_1 verbindende **Liniensegment**.

Konvexe Menge, konvexe Hülle

Sei $d \in \mathbb{N}$.

- Für $v_0, v_1 \in \mathbb{R}^d$ sei

  $$\overline{v_0 v_1} := \{(1-\lambda)v_0 + \lambda v_1 \mid \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1\} ,$$

  das v_0 und v_1 verbindende **Liniensegment**.

- Eine Menge $C \subseteq \mathbb{R}^d$ heisst **konvex**, falls

  $$\forall v_0, v_1 \in C\colon\ \overline{v_0 v_1} \subseteq C .$$

Konvexe Menge, konvexe Hülle

Sei $d \in \mathbb{N}$.

- Für $v_0, v_1 \in \mathbb{R}^d$ sei

$$\overline{v_0 v_1} := \{(1-\lambda)v_0 + \lambda v_1 \mid \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1\} \ ,$$

  das v_0 und v_1 verbindende **Liniensegment**.

- Eine Menge $C \subseteq \mathbb{R}^d$ heisst **konvex**, falls

$$\forall v_0, v_1 \in C\colon\ \overline{v_0 v_1} \subseteq C \ .$$

- Die **konvexe Hülle**, $\mathrm{conv}(S)$, einer Menge $S \subseteq \mathbb{R}^d$ ist der Schnitt aller konvexen Mengen, die S enthalten, d.h.

$$\mathrm{conv}(S) := \bigcap_{S \subseteq C \subseteq \mathbb{R}^d,\ C \text{ konvex}} C \ .$$

Konvexe Menge, konvexe Hülle

Sei $d \in \mathbb{N}$.

- Für $v_0, v_1 \in \mathbb{R}^d$ sei

  $$\overline{v_0 v_1} := \{(1-\lambda)v_0 + \lambda v_1 \mid \lambda \in \mathbb{R}, 0 \leq \lambda \leq 1\} \ ,$$

  das v_0 und v_1 verbindende **Liniensegment**.

- Eine Menge $C \subseteq \mathbb{R}^d$ heisst **konvex**, falls

  $$\forall v_0, v_1 \in C \colon \ \overline{v_0 v_1} \subseteq C \ .$$

- Die **konvexe Hülle**, $\mathrm{conv}(S)$, einer Menge $S \subseteq \mathbb{R}^d$ ist der Schnitt aller konvexen Mengen, die S enthalten, d.h.

  $$\mathrm{conv}(S) := \bigcap_{S \subseteq C \subseteq \mathbb{R}^d,\, C \text{ konvex}} C \ .$$

siehe „Diskrete Mathematik“, „Analysis“, „Lineare Algebra“.

Darstellung der konvexen Hülle

Für eine endliche Punktemenge P in der Ebene wird die konvexe Hülle durch ein Polygon, der Rand von $\text{conv}(P)$, bestimmt, dessen Ecken Punkte aus P sind. Wenn wir von der Berechnung von $\text{conv}(P)$ sprechen, so meinen wir die Bestimmung einer Folge

$$(q_0, q_1, \ldots, q_{h-1}), \quad h \leq n,$$

der Ecken dieses Polygons, beginnend bei einer beliebiger Ecke q_0 und dann entgegen dem Uhrzeigersinn entlang dieses Polygons.

Darstellung der konvexen Hülle

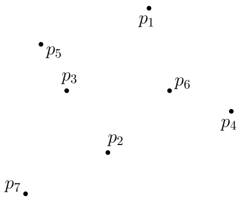
Für eine endliche Punktemenge P in der Ebene wird die konvexe Hülle durch ein Polygon, der Rand von $\mathrm{conv}(P)$, bestimmt, dessen Ecken Punkte aus P sind. Wenn wir von der Berechnung von $\mathrm{conv}(P)$ sprechen, so meinen wir die Bestimmung einer Folge

$$(q_0, q_1, \ldots, q_{h-1}), \quad h \leq n,$$

der Ecken dieses Polygons, beginnend bei einer beliebiger Ecke q_0 und dann entgegen dem Uhrzeigersinn entlang dieses Polygons.

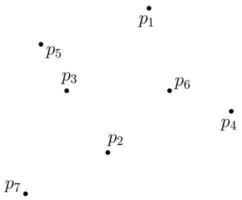
Beachte: $Q := \{q_0, q_1, \ldots, q_{h-1}\} \subseteq P$ ist die kleinste Teilmenge von P mit $\mathrm{conv}(Q) = \mathrm{conv}(P)$.

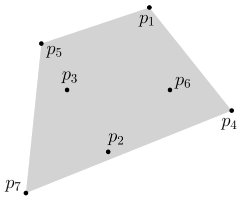
Problemstellung



Punktemenge P,

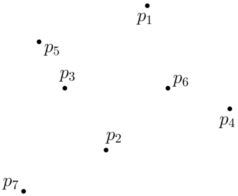
Problemstellung

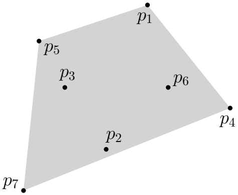


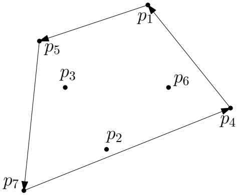


Punktemenge P, konvexe Hülle $\operatorname{conv}(P)$,

Problemstellung

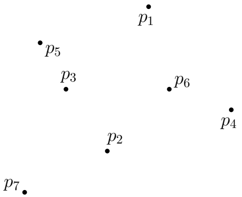




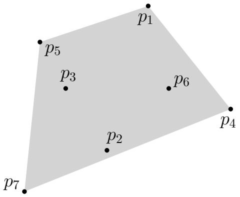


Punktemenge P, konvexe Hülle $\text{conv}(P)$, Polygon (p_4, p_1, p_5, p_7).

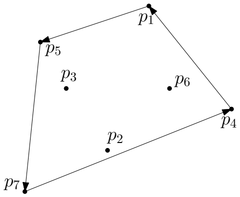
Problemstellung



Punktemenge P,



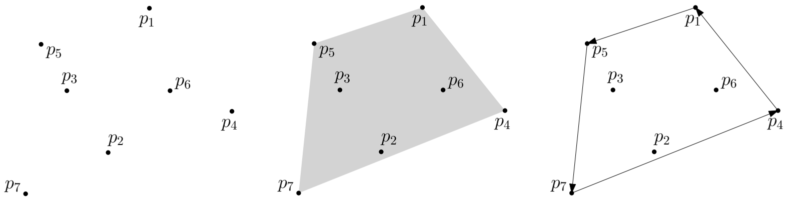
konvexe Hülle $\mathrm{conv}(P)$,



Polygon (p_4, p_1, p_5, p_7).

ConvexHull-Problem. Gegeben eine endliche Punktemenge $P \subseteq \mathbb{R}^2$, bestimme die Ecken des $\mathrm{conv}(P)$ umrandenden Polygons, in der Reihenfolge gegen den Uhrzeigersinn.

Problemstellung

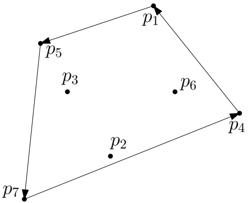


Punktemenge P, konvexe Hülle $\mathrm{conv}(P)$, Polygon (p_4, p_1, p_5, p_7).

ConvexHull-Problem. Gegeben eine endliche Punktemenge $P \subseteq \mathbb{R}^2$, bestimme die Ecken des $\mathrm{conv}(P)$ umrandenden Polygons, in der Reihenfolge gegen den Uhrzeigersinn.

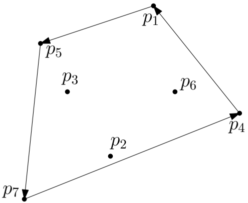
Vereinfachende Annahme: **Allgemeine Lage**, d.h. keine 3 Punkte auf einer gemeinsamen Geraden, keine 2 Pkt gleiche x-Koordinate.

Randkanten



Ein Paar $qr \in P^2$, $q \neq r$, heisst **Randkante** von P, falls alle Punkte in $P \setminus \{q, r\}$ links von qr liegen, d.h. auf der linken Seite der gerichteten Geraden durch q und r, gerichtet von q nach r, liegen.

Randkanten



Ein Paar $qr \in P^2$, $q \neq r$, heisst **Randkante** von P, falls alle Punkte in $P \setminus \{q, r\}$ links von qr liegen, d.h. auf der linken Seite der gerichteten Geraden durch q und r, gerichtet von q nach r, liegen.

Lemma

$(q_0, q_1, \ldots, q_{h-1})$ *ist die Eckenfolge des* $\mathrm{conv}(P)$ *umschliessenden Polygons gegen den Uhrzeigersinn genau dann wenn alle Paare* (q_{i-1}, q_i), $i = 1, 2, \ldots, h$, *Randkanten von* P *sind (Indizes* $\bmod h$*).*

Orientierungstest

Wie entscheiden wir „p liegt links von qr“?

Orientierungstest

Wie entscheiden wir „p liegt links von qr"?

Lemma

Seien $p = (p_x, p_y)$, $q = (q_x, q_y)$, und $r = (r_x, r_y)$ Punkte in $\mathbb{R}^2$. Es gilt $q \neq r$ und p liegt links von qr genau dann wenn

$$\begin{aligned} \det(p,q,r) &:= \begin{vmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ r_x & r_y & 1 \end{vmatrix} = \begin{vmatrix} q_x - p_x & q_y - p_y \\ r_x - p_x & r_y - p_y \end{vmatrix} > 0 \\ &\Leftrightarrow \quad (q_x - p_x)(r_y - p_y) > (q_y - p_y)(r_x - p_x) \end{aligned}$$

Orientierungstest

Wie entscheiden wir „p liegt links von qr"?

Lemma

Seien $p = (p_x, p_y)$, $q = (q_x, q_y)$, und $r = (r_x, r_y)$ Punkte in $\mathbb{R}^2$. Es gilt $q \neq r$ und p liegt links von qr genau dann wenn

$$\begin{aligned} \det(p,q,r) \quad &:= \quad \begin{vmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ r_x & r_y & 1 \end{vmatrix} = \begin{vmatrix} q_x - p_x & q_y - p_y \\ r_x - p_x & r_y - p_y \end{vmatrix} > 0 \\ \Leftrightarrow \quad & \quad (q_x - p_x)(r_y - p_y) > (q_y - p_y)(r_x - p_x) \end{aligned}$$

$$\frac{1}{2}|\det(p,q,r)| = \text{die Fläche des Dreiecks } pqr$$

Das Vorzeichen von $\det(p, q, r)$ bestimmt, wie die Punkte p, q, r den Rand dieses Dreiecks durchlaufen.

Erster naiver Ansatz

Gehe durch jedes der $n(n-1)$ geordneten Paare qr, und prüfe, ob dies eine Randkante ist,

Erster naiver Ansatz

Gehe durch jedes der $n(n-1)$ geordneten Paare qr, und prüfe, ob dies eine Randkante ist,

> indem man für alle $n-2$ Punkte p in $P \setminus \{q, r\}$ feststellt, ob p links von qr liegt.

So haben wir die Randkanten in $O(n^3)$ gefunden, die wir nur mehr richtig aneinanderreihen müssen.

Finden des nächsten Punkts

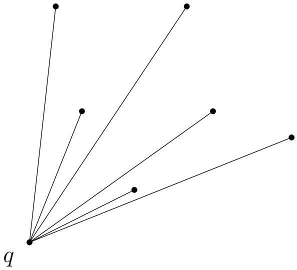
$q_0 :=$ Punkt mit kleinster x-Koordinate in P.

q_0 ist sicher eine Ecke der konvexen Hülle, also Teil der gesuchten Folge und wir können insbesondere die Folge auch mit q_0 beginnen.

Wie finden wir den Punkt q_1, der die Randkante q_0q_1 bildet?

FindNext(q)

Wähle $p_0 \in P \setminus \{q\}$ beliebig
$q_{\mathrm{next}} \leftarrow p_0$
for all $p \in P \setminus \{q, p_0\}$ **do**
    if p rechts von qq_{next} **then**
        $q_{\mathrm{next}} \leftarrow p$
return q_{next}



Ordnung um einen Punkt

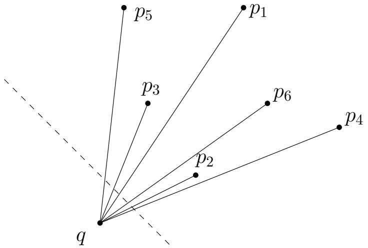
Gegeben $q \in P$, sei $\prec_q$ eine Relation auf $P \setminus \{q\}$ mittels

$$p_1 \prec_q p_2 \quad :\Leftrightarrow \quad p_1 \text{ rechts von } qp_2$$

Ordnung um einen Punkt

Gegeben $q \in P$, sei $\prec_q$ eine Relation auf $P \setminus \{q\}$ mittels

$$p_1 \prec_q p_2 \quad :\Leftrightarrow \quad p_1 \text{ rechts von } qp_2$$

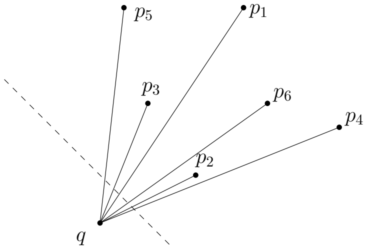


$$p_4 \prec_q p_2 \prec_q p_6 \prec_q p_1 \prec_q p_3 \prec_q p_5$$

Ordnung um einen Punkt

Gegeben $q \in P$, sei $\prec_q$ eine Relation auf $P \setminus \{q\}$ mittels

$$p_1 \prec_q p_2 \quad :\Leftrightarrow \quad p_1 \text{ rechts von } qp_2$$



$$p_4 \prec_q p_2 \prec_q p_6 \prec_q p_1 \prec_q p_3 \prec_q p_5$$

qp_4 ist Randkante.

Lemma

Ist q eine Ecke der konvexen Hülle von P, so ist die Relation $\prec_q$ eine totale Ordnung auf $P \setminus \{q\}$. Für das Minimum $p_{\min}$ dieser Ordnung gilt, dass $qp_{\min}$ eine Randkante ist.

Jarvis Wrap (Einwickeln)

JarvisWrap(P)

$h \leftarrow 0$
$p_{\mathrm{now}} \leftarrow$ Punkt in P mit kleinster x-Koordinate
repeat
  $q_h \leftarrow p_{\mathrm{now}}$
  $p_{\mathrm{now}} \leftarrow \mathrm{FindNext}(q_h)$
  $h \leftarrow h + 1$
until $p_{\mathrm{now}} = q_0$
return $(q_0, q_1, \ldots, q_{h-1})$

Jarvis Wrap (Einwickeln)

JarvisWrap(P)

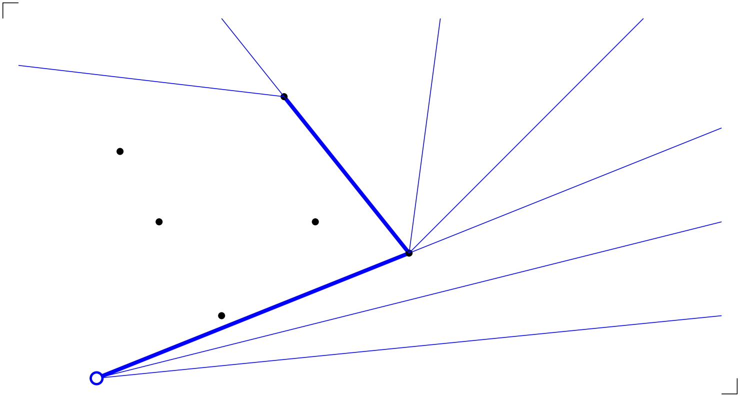
$h \leftarrow 0$
$p_{\mathrm{now}} \leftarrow$ Punkt in P mit kleinster x-Koordinate
repeat
  $q_h \leftarrow p_{\mathrm{now}}$
  $p_{\mathrm{now}} \leftarrow$ FindNext(q_h)
  $h \leftarrow h + 1$
until $p_{\mathrm{now}} = q_0$
return $(q_0, q_1, \ldots, q_{h-1})$

Satz

Für eine Menge P von n Punkten in allgemeiner Lage in $\mathbb{R}^2$ berechnet der Algorithmus JarvisWrap *die konvexe Hülle in Zeit $O(nh)$, wobei h die Anzahl der Ecken der konvexen Hülle von P ist.*

Einwickeln (wrap)

Einwickeln (wrap)



Jarvis Wrap – Laufzeit

Satz

Für eine Menge P von n Punkten in allgemeiner Lage in $\mathbb{R}^2$ berechnet der Algorithmus JarvisWrap *die konvexe Hülle in Zeit $O(nh)$, wobei h die Anzahl der Ecken der konvexen Hülle von P ist.*

Jarvis Wrap – Laufzeit

Satz

Für eine Menge P von n Punkten in allgemeiner Lage in $\mathbb{R}^2$ berechnet der Algorithmus JarvisWrap *die konvexe Hülle in Zeit $O(nh)$, wobei h die Anzahl der Ecken der konvexen Hülle von P ist.*

- Da $h \leq n$, läuft JarvisWrap in $O(n^2)$ (statt $O(n^3)$).
- Ist $h = O(1)$, z.B. $\text{conv}(P)$ ist ein Dreieck, läuft der Algorithmus in $O(n)$ Zeit.

Jarvis Wrap – Laufzeit

Satz

Für eine Menge P von n Punkten in allgemeiner Lage in $\mathbb{R}^2$ berechnet der Algorithmus JarvisWrap *die konvexe Hülle in Zeit $O(nh)$, wobei h die Anzahl der Ecken der konvexen Hülle von P ist.*

- Da $h \leq n$, läuft JarvisWrap in $O(n^2)$ (statt $O(n^3)$).
- Ist $h = O(1)$, z.B. $\mathrm{conv}(P)$ ist ein Dreieck, läuft der Algorithmus in $O(n)$ Zeit.

Für Punkte zufällig in einem Quadrat: Erwartete #Ecken $O(\log n)$.

Jarvis Wrap – Laufzeit

Satz

Für eine Menge P von n Punkten in allgemeiner Lage in $\mathbb{R}^2$ berechnet der Algorithmus JarvisWrap *die konvexe Hülle in Zeit $O(nh)$, wobei h die Anzahl der Ecken der konvexen Hülle von P ist.*

- Da $h \leq n$, läuft JarvisWrap in $O(n^2)$ (statt $O(n^3)$).
- Ist $h = O(1)$, z.B. $\mathrm{conv}(P)$ ist ein Dreieck, läuft der Algorithmus in $O(n)$ Zeit.

Für Punkte zufällig in einem Quadrat: Erwartete #Ecken $O(\log n)$.

Für Punkte zufällig in einer Kreisscheibe: Erwartete #Ecken $O(\sqrt[3]{n})$.

Implementierung – Degeneriertheiten

Kollinearitäten (3 Punkte auf Gerade), **gleiche x-Koord.**, ...

- Anfangspunkt q_0 als den Punkt mit lexikographisch kleinster Koordinate (unter allen mit kleinster x-Koordinate, den mit kleinster y-Koordinate). (Adaption der x-Ordnung der Punkte)
- Der Test "p rechts von qq_{next}" muss ersetzt werden durch (p rechts von qq_{next}) oder (p auf der Geraden durch qq_{next} und $|qp| > |qq_{\mathrm{next}}|$). (Adaption der Ordnung $\prec_q$)
- In der Regel können wir nicht einmal annehmen, dass die Punkte verschieden sind (z.B. gegeben in einem Feld)!

Software ohne Berücksichtigung dieser Fälle hat geringen Nutzen!

Implementierung – numerische Probleme

$$\text{``}(q_x - p_x)(r_y - p_y) > (q_y - p_y)(r_x - p_x)\text{''}$$

ist in einer Implementierung mit Fliesskommazahlen nicht exakt.

Implementierung – numerische Probleme

$$``(q_x - p_x)(r_y - p_y) > (q_y - p_y)(r_x - p_x)"$$

ist in einer Implementierung mit Fliesskommazahlen nicht exakt.

Es geht oft nicht um die absolute Genauigkeit des Ergebnisses (z.B. bei Eingaben, die selbst schon mit Fehlern behaftet sind). Vielmehr kann der Algorithmus völlig falsche Ergebnisse liefern oder in eine unendliche Schleife laufen.

Implementierung – numerische Probleme

$$``(q_x - p_x)(r_y - p_y) > (q_y - p_y)(r_x - p_x)''$$

ist in einer Implementierung mit Fliesskommazahlen nicht exakt.

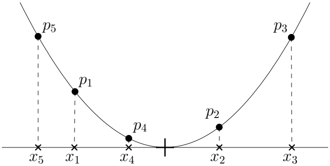
Es geht oft nicht um die absolute Genauigkeit des Ergebnisses (z.B. bei Eingaben, die selbst schon mit Fehlern behaftet sind). Vielmehr kann der Algorithmus völlig falsche Ergebnisse liefern oder in eine unendliche Schleife laufen.

Zum Beispiel **Vorbeilaufen am Startpunkt** (wegen numerischer Probleme, weil Startpunkt doppelt auftritt).

Programmbibliotheken bieten exakte Datentypen
(für spezielle Operationen).

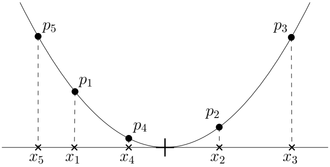
Untere Schranke für ConvexHull

Betrachte eine Folge $(x_1, x_2, \ldots, x_n)$ von Zahlen in $\mathbb{R}$. Wir setzen $p_i = (x_i, x_i^2)$, $i = 1, 2, \ldots, n$ (vertikale Projektion von der x-Achse im $\mathbb{R}^2$ auf die Einheitsparabel $y = x^2$).



Untere Schranke für ConvexHull

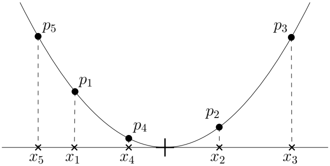
Betrachte eine Folge $(x_1, x_2, \ldots, x_n)$ von Zahlen in $\mathbb{R}$. Wir setzen $p_i = (x_i, x_i^2)$, $i = 1, 2, \ldots, n$ (vertikale Projektion von der x-Achse im $\mathbb{R}^2$ auf die Einheitsparabel $y = x^2$).



Aus der Folge der Ecken der konvexen Hülle von $P := \{p_1, p_2, \ldots, p_n\}$ ergibt sich (in linearer Zeit) auch die aufsteigend sortierte Reihenfolge der x_i's.

Untere Schranke für ConvexHull

Betrachte eine Folge $(x_1, x_2, \ldots, x_n)$ von Zahlen in $\mathbb{R}$. Wir setzen $p_i = (x_i, x_i^2)$, $i = 1, 2, \ldots, n$ (vertikale Projektion von der x-Achse im $\mathbb{R}^2$ auf die Einheitsparabel $y = x^2$).



Aus der Folge der Ecken der konvexen Hülle von $P := \{p_1, p_2, \ldots, p_n\}$ ergibt sich (in linearer Zeit) auch die aufsteigend sortierte Reihenfolge der x_i's.

Wir haben eine sogenannte **Reduktion** gezeigt: Kann man ConvexHull in $t(n)$ lösen, so kann man in $t(n) + O(n)$ Zeit sortieren.